

## **Лабораторна робота №3.**

### **Дослідження нових команд МК Mega таXmega**

**Тема:** Нові команди AVR-мікроконтролерів

**Мета:** Користуючись налагоджувачем дослідити виконання нових команд AVR-мікроконтролерів у покроковому режимі.

#### **Порядок виконання лабораторної роботи:**

- 1) Вивчити теоретичні відомості з теми «Нові команди AVR-мікроконтролерів». Дослідити формати (типи) команд, представлення операндів і роботу програми, що наведено у лабораторній роботі як приклад. Вміти коментувати нові команди, які наведено у таблиці 1.54.
- 2) Ввести програму з використанням нових команд у симулятор AVR-Studio 4.
- 3) За допомогою програмно-налагоджувальних засобів у AVR-Studio 4 проаналізувати виконання цієї програми та переконатися у правильному її виконанні. При негативному результаті здійснити зміну програми та повторити перевірку.
- 4) Роздрукувати лістинг правильно працюючої програми.
- 5) Відповісти на контрольні питання викладача.

#### **Стислі теоретичні відомості**

Команди даної групи наведено у таблиці 1.5, а їх опис – у підрозділі 1.11.

Нижче наведено програму з використанням нових команд.

#### **Програма з використанням нових команд:**

ok3:  
пор

; Порожня операція

```

пор                ; Порожня операція
; Множення
ldi r19, $ 2f      ; Завантажити константу r19 = $ 2f
ldi r20, $ 78      ; Завантажити константу r20 = $ 78
MUL r19, r20       ; Множення беззнакових чисел
; Інструкція переходу
jmp ok            ; Відносний безумовний перехід
пор                ; Порожня операція
пор                ; Порожня операція
пор                ; Порожня операція
ок:
; Множення
ldi r17, $ 2f      ; Завантажити константу r17 = $ 2f
ldi r18, $ 78      ; Завантажити константу r18 = $ 78
clr r0; обнулити r0
clr r1; обнулити r1
MULS r17, r18     ; Множення знакових чисел
ldi r17, $ 2f      ; Завантажити константу r17 = $ 2f
ldi r18, $ 78      ; Завантажити константу r18 = $ 78
clr r0;
clr r1;
MULSU r17, r18    ; Множення знакового числа на беззнакове
; Інструкція переходу
ldi r30, $ 1a      ; Завантажити константу r30 = $ 8a
ldi r31, $ 00      ; Завантажити константу r31 = $ 00
EIJMP              ; Безумовний непрямий перехід в пам'яті програм ємністю 4
; Мслова на Z
пор                ; Порожня операція
пор                ; Порожня операція
пор                ; Порожня операція
ок2:
; Множення
ldi r17, $ 2f      ; Завантажити константу r30 = $ 0d
ldi r18, $ 78      ; Завантажити константу r16 = $ 00MUL R1, R2; Множення бе
clr r0              ; обнулити r0
clr r1              ; обнулити r1
FMUL r17, r18     ; Множення дробових беззнакових чисел
ldi r17, $ 2f      ; Завантажити константу r17 = $ 2f
ldi r18, $ 78      ; Завантажити константу r18 = $ 78
clr r0              ; обнулити r0
clr r1              ; обнулити r1
FMULSU R17, R18   ; Множення дробового знакового числа на беззнакове
; Виклик підпрограми
ldi r25, $ de      ; Завантажити константу r25 = $ de
out $ 3D, r25      ; Записати r25 в порт $ 3D
call ok1           ; Відносний безумовний виклик
; Множення
ldi r17, $ 2f      ; Завантажити константу r17 = $ 2f
ldi r18, $ 78      ; Завантажити константу r18 = $ 78
clr r0              ; обнулити r0
clr r1              ; обнулити r1
FMULS r17, r18    ; Множення дробових знакових чисел
; Виклик підпрограми
ldi r30, $ 4b      ; Завантажити константу r30 = $ 8a

```

```

ldi r31, $ 00      ; Завантажити константу r31 = $ 00
eicall             ; Відносний безумовний виклик підпрограми
; Передача даних
ldi r19, $ 00      ; Завантажити константу r17 = $ 2f
ldi r18, $ 1b      ; Завантажити константу r18 = $ 78
clr r31            ; обнулити r31
clr r30            ; обнулити r30
MOVW r30, r18      ; Копіювання слова
LPM r1, Z           ; Завантаження регістра R1 з пам'яті програм з адресою в
; індексному регістрі Z
LPM r2, Z +        ; Завантаження регістра R1 з пам'яті програм з адресою в
; індексному регістрі Z з постінкрементом вмісту Z
SPM                ; Самопрограмування мікроконтролера, у якого є відповідний блок
clr r0             ; обнулити r0
ELPM               ; Завантажити регістра r0 з розширеною пам'яті програм з
; адресою в регістрах RAMPZ і Z
ELPM r3, Z         ; Завантажити регістра r3 з розширеної пам'яті програм з
; адресою в регістрах RAMPZ і Z
ELPM r4, Z +       ; Завантажити регістра r4 з розширеною пам'яті програм з адресою в
; регістрах RAMPZ і Z з подальшим інкрементом вмісту RAMPZ: Z
por                ; Порожня операція
por                ; Порожня операція
por                ; Порожня операція
; Інструкція переходу
jmp ok3            ; Відносний безумовний перехід
ok1:
ldi r19, 156        ; Завантажити константу r19 = 156
ldi r20, 175        ; Завантажити константу r20 = 175
ldi r21, 132        ; Завантажити константу r21 = 132
ldi r22, 87         ; Завантажити константу r22 = 87
por                ; Порожня операція
por                ; Порожня операція
por                ; Порожня операція
add r21, r19        ; Додати r21 до r19 без перенесення
adc r22, r20        ; Додати r22 до r20 з перенесенням
ret                ; Повернення з підпрограми
ok4:
ldi r23, $ 12       ; Завантажити константу r23 = $ 12
ldi r22, $ 15       ; Завантажити константу r22 = $ 15
ldi r21, $ 25       ; Завантажити константу r21 = $ 25
ldi r20, $ 87       ; Завантажити константу r20 = $ 87
sub r22, r20        ; Відняти r20 від r22 без перенесення
adc r23, r21        ; Відняти r21 від r23 з перенесенням
ret                ; Повернення з підпрограми

```

## Контрольні запитання

- 1) На які сімейства поділяють мікроконтролери AVR?
- 2) Які операції дозволяють виконувати нові команди, розглянуті у лабораторній роботі?
- 3) Які способи адресації використано у розглянутих командах?
- 4) Як представлені дробові числа у розглянутих командах?
- 5) Для чого використовують команду SPM, в чому її особливість?